Design and Motion Planning of an Autonomous Climbing Robot with Claws

Corresponding Author	Avishai Sintov, Department of Mechanical Engineering, Ben-Gurion University of the Negev, Beer Sheva, Israel. E-mail: sintova@bgu.ac.il Tel: +972-54-5562555 Fax: +972-77-5232651 Postal Address: P.O.Box 256, Omer, 84965, Israel.
Author	Tomer Avramovich**,** Department of Mechanical Engineering**,** Ben-Gurion University of the Negev, Beer Sheva 84105, Israel. tomera@bgu.ac.il
Author	Amir Shapiro, Department of Mechanical Engineering, Ben-Gurion University of the Negev, Beer Sheva 84105, Israel. ashapiro@bgu.ac.il

Abstract

This paper presents the design of a novel robot capable of climbing on vertical and rough surfaces, such as stucco walls. Termed CLIBO (claw inspired robot), the robot can remain in position for a long period of time. Such a capability offers important civilian and military advantages such as surveillance, observation, search and rescue and even for entertainment and games. The robot's kinematics and motion, is a combination between mimicking a technique commonly used in rock-climbing using four limbs to climb and a method used by cats to climb on trees with their claws. It uses four legs, each with four-degrees-of-freedom (4-DOF) and specially designed claws attached to each leg that enable it to maneuver itself up the wall and to move in any direction. At the tip of each leg is a gripping device made of 12 fishing hooks and aligned in such a way that each hook can move independently on the wall's surface. This design has the advantage of not requiring a tail-like structure that would press against the surface to balance its weight. A locomotion algorithm was developed to provide the robot with an autonomous capability for climbing along the pre-designed route. The algorithm takes into account the kinematics of the robot and the contact forces applied on the foot pads. In addition, the design provides the robot with the ability to review its gripping strength in order to achieve and maintain a high degree of reliability in its attachment to the wall. An experimental robot was built to validate the model and its motion algorithm. Experiments demonstrate the high reliability of the special gripping device and the efficiency of the motion planning algorithm.

Keywords

Climbing, Robot, Claws, Motion, Algorithm.

1. Introduction

This paper considers the design and motion planning of a robot with the ability to climb on vertical surfaces. Such a capability significantly increases robot mobility and workspace and has important military and civilian advantages. As part of the design goals, it was posited that the robot should be able to move in an autonomous and reliable way. Moreover, the robot should be small, compact and easy-to-carry for one-man operation. To conduct its missions, the robot must also be able to remain statically attached to the wall with no energy consumption. To achieve these design goals, a robot was designed and developed that mimics the kinematics of a human rock climber who uses four limbs to climb and implements the method used by cats to climb on trees utilizing their claws. The robot that was designed is termed CLIBO (claw inspired robot)**.**

A robot prototype was constructed for the purpose of demonstrating our concept. Using a kinematics model, the locomotion algorithm that was developed as part of this work combines control of the four legs with an ability to utilize smart actuators. Our experimental results with CLIBO have shown that reliable wall-climbing is feasible. The unique design of the robot provides it with maneuvering capabilities, on the one hand, and the ability to control its position and force distribution, on the other.

A robot that can vertically and autonomously move vertically along a rough surface such as stucco, offers considerable military and civilian advantages. Positioned high on a building, the robot, serving as an observation platform, could provide valuable military intelligence as well as assist in search and rescue operations. Such a robot could also be used for unmanned sweeps of hostile areas and serve as a platform for carrying firearms and explosives. In terms of civilian use, the robot could be used in construction to signal back the progress or state of various operations being implemented at dangerously high levels.

There are several types of robots with the capability to climb on various surfaces. By using adhesive wheel-legs for locomotion, the Mini-Whegs [1], a small quadruped robot, is able to climb on smooth vertical surfaces. The Stickybot [2] robot imitates the locomotion of a lizard and can climb on flat and smooth walls. It climbs using directional dry adhesive on its specially designed legs. There are many more wall-climbing robots using adhesive methods such as the Geckobot [3], Waalbot [4] and a miniature robot that uses a biomimetic adhesive [5]. Unlike adhesive attachment methods, the Clarifying Climber III [6] robot which uses vortex technology and the climbing robot [7] which uses Bernoulli effect, have the advantage of adhesion forces largely independent of the type of material and surface conditions. However, during the entire climbing process, the robots consume so much energy that their time of operation is limited. Other climbing robots using suction are the ROBICEN [8], NINJA-II [9], ROBIN [10], a climbing robot for inspecting nuclear power plants [11], a robot using tracked wheel mechanism with suction pads [12] and two other four limbed robots with suction pads attached [13],[14]. The suction method has problem of unreliable sealing when climbing on uneven surfaces. Moreover, suction method demands high energy consumption while attached. Magnetic attachment is another climbing method used by several robots such as [15], [16], [17] and [18] which uses permanent magnetic wheels or tracks. The robot described in [19] has and electromagnetic feet for high grasping force and permanent magnetic feet ensuring

safety in case of power break down. Magnetic robots have advantage of high climbing payloads. However, they are limited for climbing on only ferromagnetic walls. The LEMUR II [20] can autonomously climb on vertical rock–like surfaces using four limbs. LEMUR climbs on a rock climbing training wall, where the foothold locations are constrained to a set of discrete points. The RiSE [21] mimics a movement of six-legged insects. The RiSE robot uses compliant microspines on its feet for reliable attachment to rough surfaces. However Rise uses about 20 (three motors in each of the six legs, one in the middle and one at the tail) motors for maneuvering over large obstacles. On the other hand, CLIBO has a structure which provides high maneuverability and ability to transfer (although not implemented yet) between angled surfaces using only 16 motors. Another robot which is used for climbing on rough surfaces is the ROCR [22]. ROCR is a pendular two-link, serial chain robot that utilizes alternating handholds and an actuated tail to propel itself upward in a climbing style derived from observations of human climbers. These all mentioned robots have problems such as incapability to climb rough surfaces, large energy consumption and maneuvering limitations. Opposed to these robots, CLIBO's design and motion planning enables it to climb and maneuver on problematic surfaces and to remain static for a long period of time.

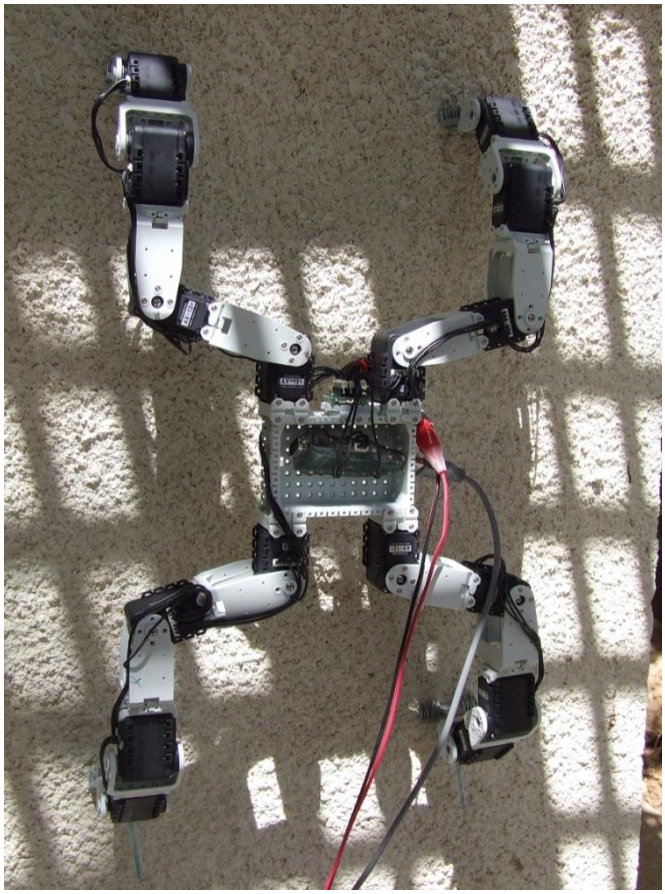


Figure 1: CLIBO prototype climbing a wall.

The first part of this paper presents a review of the consideration in the robot's design that led to its kinematic structure. In the second part we review the mathematical model of the robot, describing the kinematics and static model derived from its design. In Section 3 we discuss a motion-planning algorithm based on grip quality measures and robot kinematics. Section 4 presents the implementation of the design and the

motion planning algorithm. We also present here the prototype robot that has been built and discuss various wall-climbing experiments that were carried out with the prototype.

2. Robot Design and Analysis

In order to achieve a working robot capable of climbing rough surfaces, CLIBO's structure was developed in such a way that when activated it would mimic a rock climbing technique of climbing using four limbs. This section reviews the robot's design, its physical structure and the kinematic and static models.

2.1 Robot Design

The robot consists of four legs which are arranged symmetrically around the robot's central body. Each leg has five-degrees-of-freedom (DOF). Figure 2 describes the design of a leg. Four of the DOFs are motorized and the fifth, which is in the gripping device mounted on the tip of the leg, is a passive DOF. The first two DOFs, whose axes are perpendicular to the wall, enable the robot to move forward. These two DOFs are also responsible for controlling the attachment of the claws to the wall by pulling the end-effectors (EE), described below, down toward the floor and checking the reaction forces. The two remaining motorized DOFs whose axes are parallel to the wall's plane are designed for determining the distance of the robot from the wall (Motor 3) and the angular constraint for the EE (Motor 4).

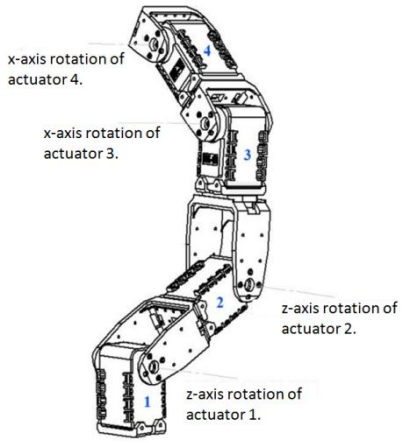


Figure 2: The leg's active 4 DOF structure.

This design of the leg provides the robot with good gait capability. The first two motors in each leg drive the robot's movement. After the attachment of the hooks and upon determination of the distance from the wall (by motors 3 and 4) of every leg, the robot's movement is made by the first two motors in each leg. This movement is similar to the movement of rock-climbers who use their fingers to grasp cracks in a rock face and activate their shoulder and elbow muscles to advance. The structure of the robot allows it to move in any desired direction (360˚), by just moving 8 of its 16 motors. Furthermore, the robot can change its distance from the wall by extending its legs, to lower or raise itself in relation to the wall's surface according to the surface

condition. Consequently, this leg design has the advantage of decoupling motion in plane (parallel to the wall) and normal to the plane.

An alternative leg configuration was examined. One in which the first DOF's axis is perpendicular to the surface and the other 3 DOF's axis are parallel to the wall's surface. Such configuration gives advantage in climbing payload and lateral movement. However, this configuration bounds the robot to operate all 4 motors while advancing. Moreover, due to the motors arrangement, the robots center of mass is shifted away from the wall and therefore acting to detach it.

Four actuators per leg were assembled with an EE at the tip of every leg. The EE gripping device (Figure 3), which imitates the way cats hold objects or surfaces when climbing, is a unique device designed especially for the robot's movement. Each device, consisting of 12 fishing hooks from nickel and aligned on an aluminum body, is capable of grasping cracks in the wall and holding up to 2 kg of weight. The hooks are connected to the aluminum body by a thin nylon string. A small piece box-formed epoxy glues the hooks to the string. Between the hooks are guides that prevents them from becoming entangled one with another and limits the epoxy piece to one passive compliant DOF. In other words, the hook is not able to move laterally or to twist. It can only move in the direction of the wall, back and forward. Experiments done on a series of hooks trying to grip simultaneously have shown that the hooks constrained to each other interfere and lack of gripping ability. This arrangement provides each hook with an independent gripping capability. The gripping device is designed in such way that the hooks are rotated at a 20˚ angle in relation to a wall's plane. Such rotation prevents the gripping device's body from colliding with the wall.

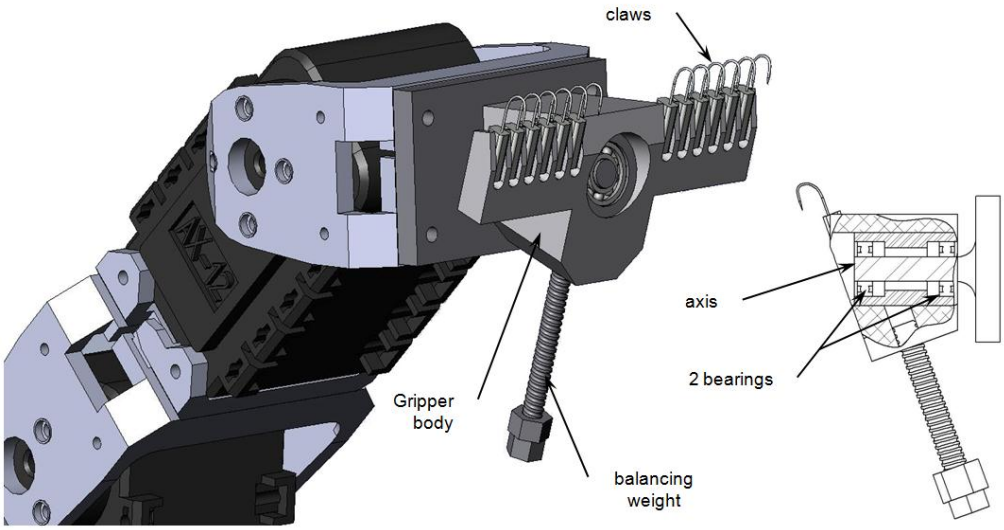


Figure 3: A scheme of the gripping device (left) and its cross-section (right).

We can roughly confine the parameters of a stucco wall for which the robot can climb on according to the claws geometry and its constraints on the EE. By surface common definitions described in [23], we can characterize the amplitude parameters of the surface based on deviation of the roughness profile from the mean line. The RMS roughness R_q of the surface been climbed is 141 μm and average roughness R_a of 83 μm. With the tested claws used and with approach angle θ_a (Figure 4) of approximately 45° of the claws, the assumed minimum value of the surface normal angle θ_{min} with respect to a horizontal line is about 40°. With these parameters, there

is a high probability of engaging at least a few asperities per centimeter of stroke. These parameters can be improved by replacing the claws with more smaller and sharp ones, while compromising with payload.

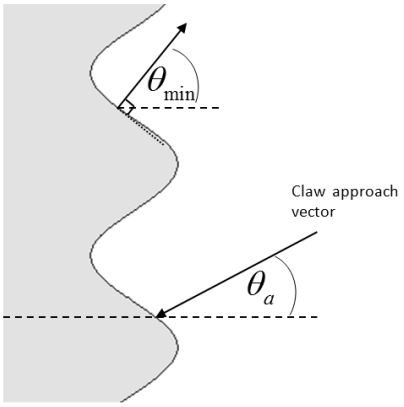


Figure 4: Angle parameters for the surface texture.

Since all the legs are fixed to the wall, the orientation of the legs must change as the robot moves its central body. The hooks attach to the wall and a change in orientation will apply torque on the gripping device about the axis perpendicular to the wall. This torque can cause the leg to disengage from the wall. In order to prevent this, a passive DOF was added to the gripping device's axis. Thus, the gripping device is attached to the leg by two miniature bearings, creating a 1 DOF axis. A small balancing weight was added to the gripping device in order to keep it horizontal as it approached the wall seeking to attach itself.

2.2 Kinematics

The first step in designing the robot's motion was to analyze its kinematics. Thus, a systematically analytical method was needed for acquiring the robot's orientation data based on the position feedbacks obtained from the servo motors.

2.2.1 Direct Kinematics

The use of direct kinematics makes it possible to pinpoint the position of the leg EEs as a function of the leg joint angles. Based on the joint angles, the EE positions can be calculated in relation to the global frame. In order to analyze the kinematics, a set of frames is attached to the system (Figure 5). The robot moves relative to Frame *W*, the global frame. Frame *0*, positioned on the robot's central body, keeps its parallelism to frame *W*. Frame *B* is fixed to the robot's central body. Frame *L*, fixed on the first motor of every leg, keeps its parallelism to frame *B*. Frames *i* (*i=1,2,3,4*) are frames placed on motor i's axis and rotates with it. It is assumed that the robot moves in a plane parallel to the wall.

Since all legs are similar, although in a mirror view, the position of the EE is first located in relation to the position of the first motor (frame *L*). It is then transformed into the central body frame *B*. When the leg is fully stretched sideways, all the angles are set to zero.

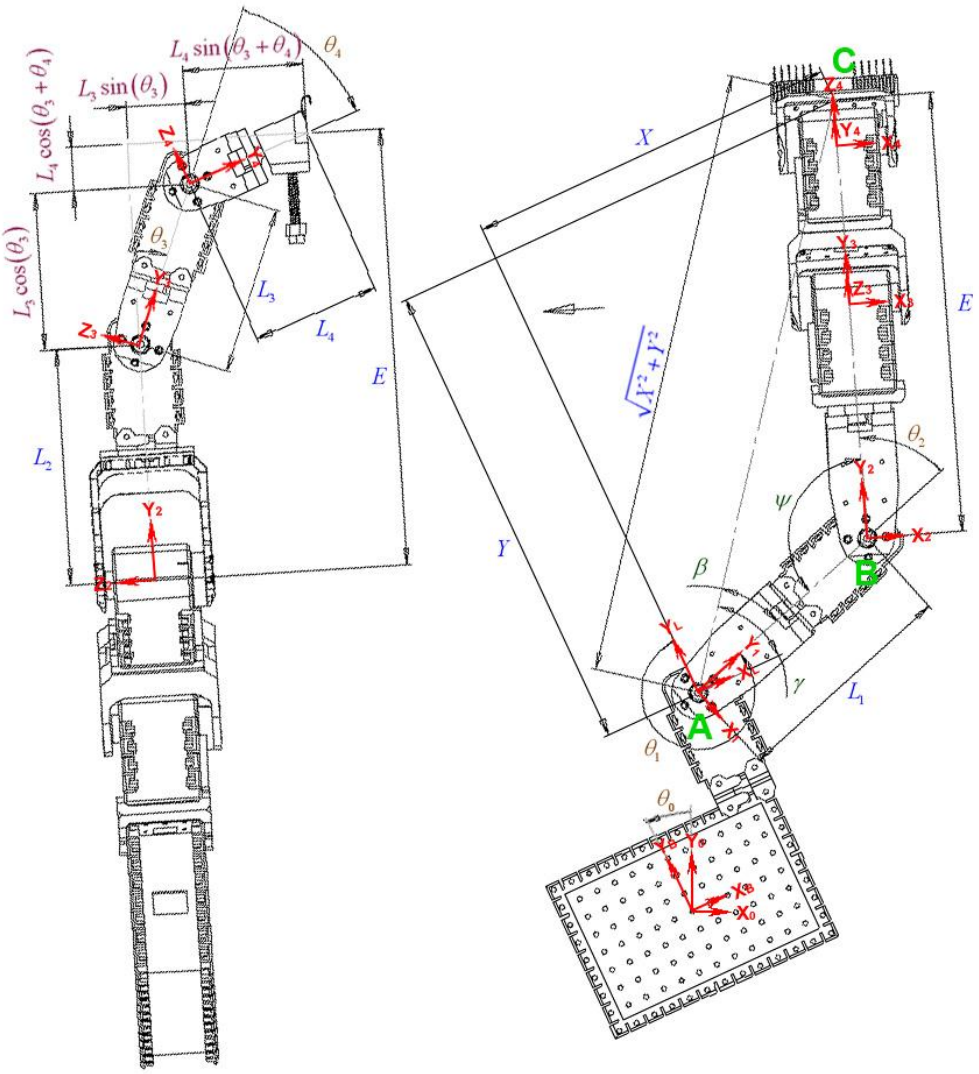


Figure 5: Coordinate frames attached to a leg.

Let frame *4* be the EE frame. The vector $\boldsymbol{r_L}$ which expresses the position of the EE position at frame *L* is:

$$\boldsymbol{r_L} = A_1^L \cdot A_1^2 \cdot A_2^3 \cdot A_3^4 \cdot \boldsymbol{r_4} \tag{1}$$

Where A_i^j is a homogenous transformation matrix from frame *i* to frame *j*, $\boldsymbol{r_4}$ is the position of the EE related to frame *4*.

Hence, the EE position with respect to the frame *L*:

$$\boldsymbol{r_L} = \begin{pmatrix} -\sin(\theta_1+\theta_2)\left[L_4\cos(\theta_3+\theta_4)+L_3\cos(\theta_3)+L_2\right]-\sin(\theta_1)L_1 \\ \cos(\theta_1+\theta_2)\left[L_4\cos(\theta_3+\theta_4)+L_3\cos(\theta_3)+L_2\right]+\cos(\theta_1)L_1 \\ L_3\sin(\theta_3)+L_4\cos(\theta_3+\theta_4) \end{pmatrix} \tag{2}$$

Where L_i is the length of the i'th link, θ_i is the angle between link *i-1* and link *i*.

As there are four legs mirrored at each side, then for every leg, $\boldsymbol{r_L}$ is mapped to frame *B* and is expressed by the vector $\boldsymbol{r_B}$:

$$\boldsymbol{r_B} = \mathrm{A}_L^B \cdot \boldsymbol{r_L} \tag{3}$$

A_L^B is the homogenous transformation matrix from frame *L* to frame *B* where rotations by φ_{By} around *x* axis and by φ_{Bx} around *y* axis are made. Each leg's constants, φ_{Bx} and φ_{By} , are given by the position of the leg around the central body and can be either 0° or 180°.

We use frame *B* to represent the position of the leg EEs in relation to the current central body position of the robot. However, because the legs are similar, all the legs movements will be controlled in frame *L* by the same global function.

2.2.2 Inverse Kinematics

To position the EE at a desired location, we use the inverse kinematics (IK) which defines the legs' matching angles. This means that a certain configuration would give the desired position of the leg EEs. The IK is used for a single leg relative to CLIBO's central body. The IK is used to reach a desired EEs position in relative to the central body according to the legs' matching angles. The orientation of the EE remains constant due to the balancing weight mentioned previously. The calculation of the IK is made with the assumption that the central body's orientation remains vertical at all time. This assumption is accurate due to orientation angle correction of the central body which will be made within the motion algorithm, as will be detailed later. Moreover, the distance of the central body from the wall is constrained to the defined value Z. By construction of the leg, the two lateral joints are responsible for the distance from the wall and the approach angle of the claws. While the two joints which are closer to the central base are responsible for the location of the contact point in the X-Y plan. With these constraints and assumptions, there are four different solutions for the desired angles, two for θ_1, θ_2 and two for θ_3, θ_4. Therefore, as we search for the same configuration solution for all legs, frame *L* of every leg is fixed as a mirror view to its neighbor. Denoting the position of the EE as $(X, Y)^T$, the inverse kinematics, computed using the leg geometrics, is calculated in frame *L*. Let variable *E* be the projection of the distance from frame *2*'s origin to the EE on the global frame's *x*-*y* plane (Figure 5). From the law of cosines, θ_2 is:

$$\theta_2 = 180 - \arccos\left(\frac{{L_1}^2 + E^2 - \left(X^2 + Y^2\right)}{2L_1E}\right) \tag{4}$$

From the law of sines, θ_1 will be:

$$\theta_1 = 270 + \arctan\frac{Y}{X} - \arcsin\left(\frac{E\sin(\psi)}{\sqrt{X^2 + Y^2}}\right) \tag{5}$$

When the leg is attached to the wall, the distance from the wall *Z* remains constant. Therefore, the sum of θ_3 and θ_4 which defines the distance *Z*, remains constant and is given by κ. From (2):

$$Z = L_3\sin(\theta_3) + L_4\sin(\kappa) \tag{6}$$

From (6), we can extract θ_3:

$$\theta_3 = \arcsin\left(\frac{Z - L_4\sin(\kappa)}{L_3}\right) \tag{7}$$

Therefore, θ_4 will be:

$$\theta_4 = \kappa - \theta_3 \tag{8}$$

This method is used in real-time for determining what the joint angles are in order to position the leg EEs at a desired position. Once the distance *Z* has been determined by the user's interface according to environment, the *4* angles can subsequently be calculated by (4)-(8).

2.3 Equilibrium analysis

The legs are composed of smart servo motors able to measure torque operating on the leg joints. Using this feedback we can calculate the force acting on the EE based on the torque of the joints. The force calculation contains the gravitation force acting at the links' center of mass. The data from determining the reaction forces acting on the leg EEs indicates one of two states. Large forces indicate that the legs are overloaded. This is dangerous for the robot's stability and needs to be dealt with immediately. Small forces can indicate that a leg has been detached from the wall.

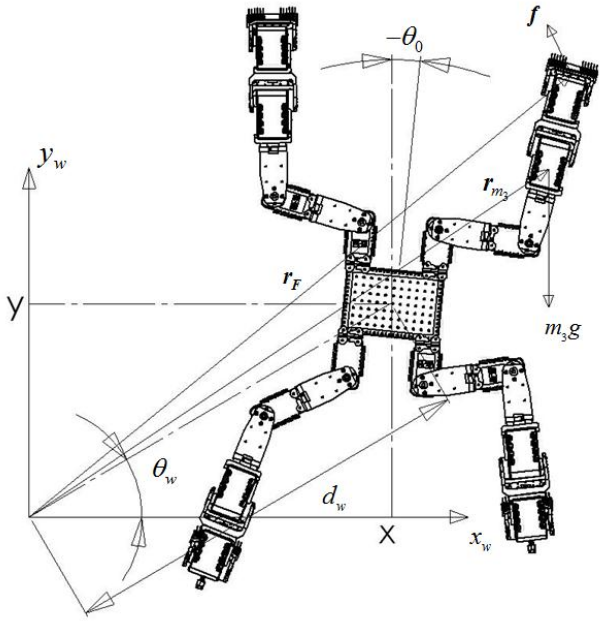


Figure 6: Forces and robots DOF parameters.

The configuration parameters vector $\boldsymbol{v_p}$, which consists of the four joint angles of the actuators $\theta_1,..,\theta_4$, the orientation angle of the central body θ_0 and its global position θ_w, d_W can be defined as follows:

$$\boldsymbol{v_P} = \begin{bmatrix} \theta_w & d_w & \theta_0 & \theta_1 & \theta_2 & \theta_3 & \theta_4 \end{bmatrix}^T \tag{9}$$

Where θ_w and d_W (Figure 6) are the position parameters of the robot related to the global frame and are given by

$$\theta_w = tg^{-1}\left(\frac{y}{x}\right), \qquad d_w = \sqrt{x^2 + y^2} \tag{10}$$

let $\boldsymbol{r_F}$ be the vector from the origin of the global frame to the EE. The EE force Jacobian will then be:

$$J_F = \frac{\partial \boldsymbol{r}_{\boldsymbol{F}}}{\partial \boldsymbol{v}_{\boldsymbol{p}}} \tag{11}$$

and the gravitational forces Jacobian:

$$J_F = \frac{\partial \boldsymbol{r}_{\boldsymbol{m}_i}}{\partial \boldsymbol{v}_{\boldsymbol{p}}} \tag{12}$$

where $\boldsymbol{r}_{\boldsymbol{m}_i} = (r_{m_i,x}, r_{m_i,y}, r_{m_i,z})^T$ is the vector from the origin of the global frame to the i'th link center of mass.

For each leg, the torques, acting on joints $\theta_1,..,\theta_4$ and on the central body θ_w, d_w, θ_0 due to the reaction force $\boldsymbol{f}$ and the links mass's m_i, are:

$$\boldsymbol{M}_{\boldsymbol{Tot}} = J_F{}^T \cdot \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} + J_{m_4}{}^T \cdot \begin{pmatrix} 0 \\ -m_4 g \\ 0 \end{pmatrix} + J_{m_3}{}^T \cdot \begin{pmatrix} 0 \\ -m_3 g \\ 0 \end{pmatrix} + J_{m_2}{}^T \cdot \begin{pmatrix} 0 \\ -m_2 g \\ 0 \end{pmatrix} + J_{m_1}{}^T \cdot \begin{pmatrix} 0 \\ -m_1 g \\ 0 \end{pmatrix} + J_{m_B}{}^T \cdot \begin{pmatrix} 0 \\ -m_B g \\ 0 \end{pmatrix} \tag{13}$$

where (f_x, f_y, f_z) are the force vectors acting on the leg's EE.

We have received a vector of torques (and one force F_w):

$$\boldsymbol{M}_{\boldsymbol{Tot}} = \begin{pmatrix} M_w & F_w & M_0 & M_1 & M_2 & M_3 & M_4 \end{pmatrix}^T, \tag{14}$$

where M_w, F_w, M_0 are torques and force acting on the central body. However, these parameters have no significance in our case. The other four parameters $M_1,..,M_4$ are the torques acting on the joints of the legs. These parameters are measured by the servo motors. Therefore, we obtained four equations:

$$\boldsymbol{M}_{\boldsymbol{Tot}} = \begin{pmatrix} M_1(f_x, f_y, f_z, \theta_0) \\ M_2(f_x, f_y, f_z, \theta_0) \\ M_3(f_x, f_y, f_z, \theta_0) \\ M_4(f_x, f_y, f_z, \theta_0) \end{pmatrix} \tag{15}$$

These four equations are the torques of the joints which features four unknown parameters f_x, f_y, f_z, θ_0. These equations are solved numerically to obtain the contact force. As expected, the solution shows that the expressions for $M_1,..,M_4$ are independent of θ_w, d_w. This means that the joints' torques do not depend on the position of the robot on the wall.

The control program can now solve these four equations in real time at any given position of the robot, providing us with information about the forces operating on the robot. The reaction force analysis is performed for one leg at a time. The analyses are compared one to another in order to analyze the weight distribution on the robot's legs.

3. Motion Planning

In this section we describe CLIBO's motion-planning algorithm which allows it to climb vertical, rough textured walls. The motion planning is based on the ability of the motors to measure the applied torque and therefore to estimate the contact force at the gripper. CLIBO's control is based on active position control and not on active force control. This way, torques and force equilibrium are obtained passively. Active force control is not feasible with our hardware, due to torque error readings from the actuators internal torque sensor and active force control under such errors may cause

loss of stability. Therefore, equilibrium is not checked. Instead, applied torques and contact forces are calculated constantly for each leg separately during the motion. The main assumption of the robots' motion is that a leg will keep trying to get a grab on the wall and eventually succeed. If succeeding to grab only after multiple tries, there can be some central body configurations which will not be feasible. There is no a priori knowledge of the surface texture, hence this assumption is inevitable.

3.1 Locomotion Principle

The principle of CLIBO's locomotion is based on the motion of the central body along a given path. The path for the central body is predefined by the user prior to climb. There is no prior knowledge of the surface to be climbed other than its perpendicularity, therefore, the footholds position are decided on-line while climbing.
The path given by the user prior to climb is discretized into small segments. The robot moves its central body towards a temporary position in a segment of its path while searching for opportunities to move its legs. Figure 7 shows the flow chart of the locomotion algorithm.

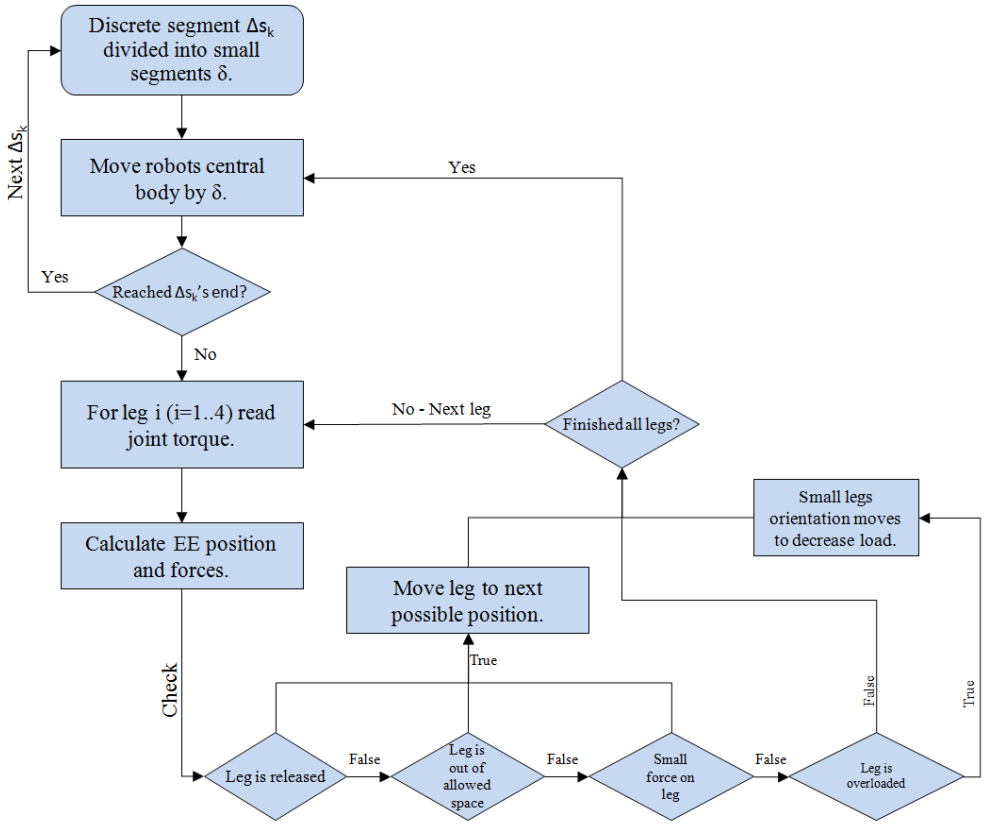


Figure 7: Flow chart for the robots movement algorithm.

The robot receives from a higher level planner a path on the wall. We wish to move the robot's central body along the given parametric path $S(\rho)$: $\mathbb{R} \longrightarrow \mathbb{R}^2$, where the parameter $\rho \in [0, \Gamma]$ is such that the Γ is maximal at the end of the path. Let $\Delta\rho$ be a path increment in the robot's path is a step of the body center. We discretize the path into $k_m = \left\lfloor \frac{\Gamma}{\Delta\rho} \right\rfloor$ elements. Hence, the k^{th} discrete point along the path is $s_k = S_k(k\Delta\rho)$. Denote $\Delta s_k = s_{k+1} - s_k$ as a discrete path element where $s_{k+1} = S((k+1)\Delta\rho)$. Every increment is then sub-divided into smaller segments with

length δ to be executed by body movements. Hence, every increment in the robots path is a Δs_k step divided into smaller, δ, sub-steps conducted by body movements.

For every increment Δs_k, the central body of the robot moves in δ steps along the linear line created by the start and the end of the increment. The motion of the central body is done by leaving the contact points in their current location and moving the central-body in a coordinated fashion using the closed chain kinematics. After every sub-step δ, torque and angle are measured at the actuators. Using inverse kinematics and static analysis (Section 2) we obtain the robot's EE positions and the forces that act on them.

The motion-planning algorithm is a reactive algorithm which continuously checks for the following four states. In each state the robot reacts differently. The robot takes an action if one of the legs is released (state 1); a leg's EE is positioned out of its allowed space (state 2); a leg load is too low (state 3); or a leg is overloaded (state 4). If none of these states occurs, the robot moves its central body to the next sub-step, repeatedly performing this 4-state check at every instance. This process continues until the robot reaches the end of the present increment Δs_k. The four states are arranged in a critical order, the most critical condition being checked first. Therefore, if state 1 (leg disengagement) is true, it is corrected before the other states are checked.

As noted, four states are routinely-checked to determine the status of the legs. State 1 is associated with the possibility of disengaging the gripping device from the wall. If such an event occurs, only small gravitational forces would act on the device, resulting in a measurement of small torques at the actuators. If this is the case, the robot would search for a new gripping point on the wall at the next possible position. The "next possible position" is a point within a leg's allowed space located along the leg's path vector. For every position of the leg, a leg's path vector is defined, starting from its current position and pointing to its final position. Its final position will be at the end point of the current segment Δs_k shifted to the leg's anticipated point next to its central body (the central body position will be in the end point of the current Δs_k).

State 2 relates to the position of the legs after the movement of the central body. The central body's movement toward its destination will increase the distance from that point to some of the leg EEs and will decrease the distance in the others. In other words, the legs need to be advanced in the path direction during the central body's δ steps. Therefore, an allowed space for each leg is defined. The allowed space specifies an area related to the central body in such a way that if the leg is located outside of this area, an action should be taken to move the leg into its allowed space along the leg's path vector. Therefore, because the allowed space is related to the central body, the movements of the central body would move the allowed space relative to the leg and would definitely cause the leg to exit it, thus forcing the robot to advance the legs in the direction of the movement. Leg advancement and allowed space definition are described in more detail in sub-section 3.2.

Relocating the central body moves the robot's center of mass along the path direction, resulting in a change of contact force distribution. Small forces acting on one leg or more may cause them to be ineffective. Moreover, small forces on some legs can cause extreme and unwanted large forces on the others. Due to these

reasons, it is essential to check states 3 and 4. State 4 is associated with a situation where a leg supports too large a force. Because actuators have a limited torque, overload on the leg is being checked per actuator. Therefore, the condition of state 4 is to check whether a torque at each leg's actuators is larger than a predefined maximal torque T_{max}. In state of overload in an actuator, the robot moves its central body away from the overloaded leg in very small steps to revise the leg's orientation and balance the torques of the actuators. As opposed to state 4, in state 3 it is necessary to check if the force of the leg acting on its EE is smaller than a defined force F_{min}. In such a case, the robot will advance the leg to the next possible position toward the end of the current segment Δs_k.

3.2 Leg Motion-Planning Algorithm

The basic principle of moving leg *i* is by calculating the next possible position of the leg with consideration of the robot's progress direction along the path. As described at sub-section 3.1, the leg has its' defined allowed space (Figure 8). The allowed space is relative to the central base and its origin is defined as the origin of frame *B*. We define R_{max} as the radius of the allowed zone and is calculated (16) as the longest possible length (*E* is defined and constant while the distance from the wall *Z* is fixed) of a leg in *x*-*y* plane divided by *SF*. *SF* is a predefined safety factor used to prevent straitening of the leg.

$$R_{\max} = \frac{L_1 + E}{SF} \tag{16}$$

Therefore, R_{max} defines y_{max} and x_{max} of the legs EE. y_{min} and x_{min} are constants and defined by the physical workspace of the leg. From the origin, two guide lines are drawn to the intersection of the arc created by R_{max} and the minimum limits (y_{min} and x_{min}) creating points *a* and *b*. This geometry generates 5 zones. The allowed zone is zone VI. After the movement of the central body, the position of the leg is checked and if it exceeds zone VI, the leg will be moved to the next possible location within the allowed zone VI. The next position will be determined according to its final destination for the current increment Δs_k. These coordinates are then processed such that the next leg's position will be in that direction but in the leg's defined allowed space. Meaning, if the next leg's desired position (x_{leg}', y_{leg}') is in zone V, it will be corrected and repositioned on the point (x_{leg}, y_{leg}) on the arc (created by R_{max}) intersecting with the movement path , if the next leg's desired position is in zone I or zone II it will be corrected and repositioned on point *a* or *b*, respectively. If the next desired position is in zone III, only the x_{leg} coordinate will be changed to be x_{min}. The same for zone IV, only the y_{leg} coordinate will be changed to be y_{min}.

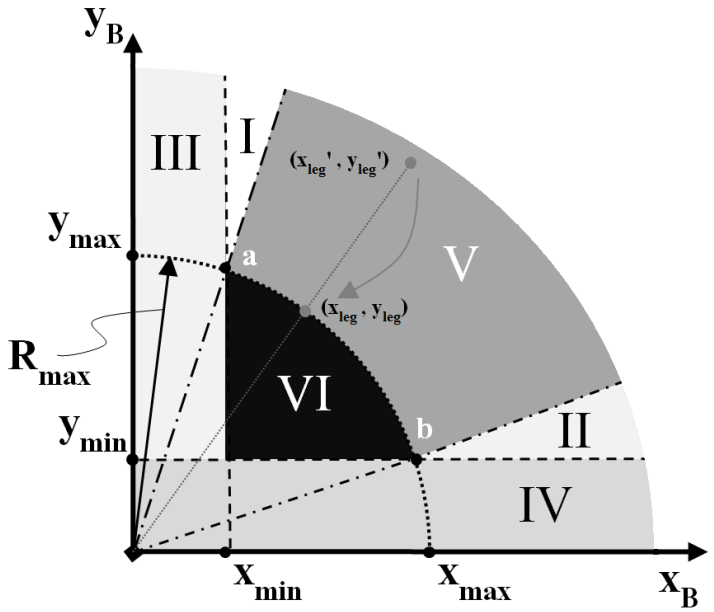


Figure 8: Allowed area in the wall plane for the robots foothold location.

Let $\Delta x_{body}, \Delta y_{body}$ be the position of the central body relative to the beginning of the current Δs_k, $x_{leg,body}, y_{leg,body}$ be the position of the leg in the end point of the current segment Δs_k shifted to the legs anticipated point next to its central body. F_{free} is minimum force acting on the EE indicating the freeing of the leg. Algorithm 1 is a sequence of actions that moves the leg *i*'s EE from its current position to the desired position $x_{leg,body}, y_{leg,body}$. $z_{leg,body}$ is a predefined constant calculated according to the defined distance from the wall Z. First the robot will detach the claws from the wall by moving it in y_w direction (toward the sky) with predefined distance *d*. Then it will measure the torques on the motors and calculate the contact force on the leg F_{leg}^{i}. If the force is smaller than F_{free}, the leg is free; else it will repeat the movement in distance *d* and check the forces again. Δx, Δy are the projections of current Δs_k on *x*,*y* axis. The distance left for the central body to reach the end of Δs_k $\Delta x - \Delta x_{body}$, $\Delta y - \Delta y_{body}$ represented in central body frame (frame B) are transformed using the homogenous transformation matrix $R_B^{L_i}$ form frame *B* to frame *L*. $R_B^{L_i}$ is given by:

$$R_B^{L_i} = \begin{pmatrix} \cos(\varphi_{By}) & 0 & -\sin(\varphi_{By}) \\ 0 & 1 & 0 \\ \sin(\varphi_{By}) & 0 & \cos(\varphi_{By}) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi_{Bx}) & -\sin(\varphi_{Bx}) \\ 0 & \sin(\varphi_{Bx}) & \cos(\varphi_{Bx}) \end{pmatrix} \qquad (17)$$

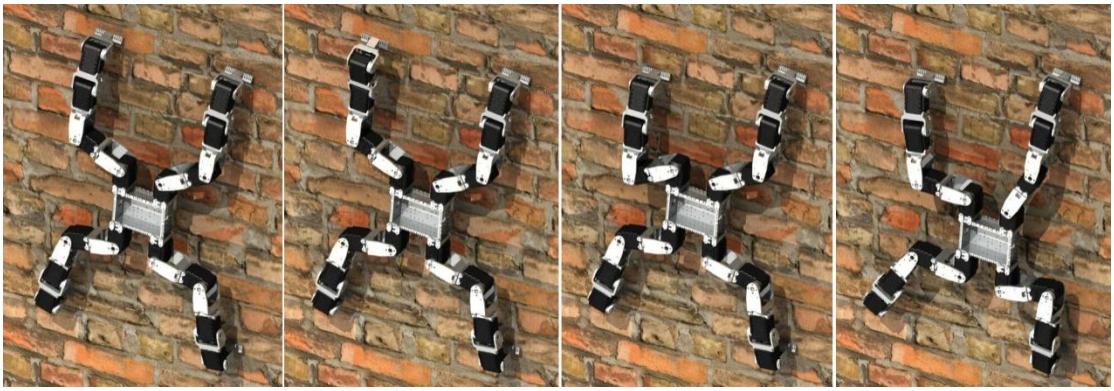


Figure 9: Illustration of the legs motion algorithm. From left to right: (1) Initial position. (2) The robot releases its upper left leg from the wall. (3) The leg is repositioned on the wall. (4) The robot moves its central body in direction of progress (down-right).

The anticipated position of the leg in the end of the current Δs_k, is added to the legs relative position to the central body. We calculated the vector from the legs current position to the legs desired position in the end of the current Δs_k expressed in the legs frame. Therefore, the final position of the leg be checked to be in the allowed space, if not it will be corrected to it as described previously.

Algorithm 1 Leg's motion planning algorithm

Input: Leg *i* to be moved and the desired position of the leg $x_{leg,body}, y_{leg,body}$.

Output: Movement of leg *i* to the nearest possible position to the desired position.

1: Let F_{free}, d = *predefined*, $F_{leg}^{i} = 0$,

2: **While** $F_{leg}^{i} > F_{free}$ **do** /*while leg *i* is still connected.

3: Move leg in y_w direction with distance d. /*disconnect claws from wall.

4: Measure motors torques.

5: Calculate F_{leg}^{i}.

6: **End while**

7: Calculate: $\begin{pmatrix} x_{leg} \\ y_{leg} \\ z_{sign} \end{pmatrix} = \begin{pmatrix} x_{leg,body} \\ y_{leg,body} \\ 0 \end{pmatrix} + R_B^{L_i} \begin{pmatrix} \Delta x - \Delta x_{body} \\ \Delta y - \Delta y_{body} \\ 1 \end{pmatrix}$, $z_{leg} = z_{leg,body} \cdot z_{sign}$

8: If $\left(x_{leg} < x_{\min}\right)$

9: Set $x_{leg} = x_{\min}$ /* the desired coordinates are in zone III.

10: If $\left(y_{leg} < y_{\min}\right)$

11: Set $y_{leg} = y_{\min}$ /* the desired coordinates are in zone IV.

12: If $\left({x_{leg}}^2 + {y_{leg}}^2 > {R_{\max}}^2\right)$

13: If $\left(\mathrm{y_{leg}} \geq \frac{y_{\max}}{\mathrm{x_{min}}} \cdot \mathrm{x_{leg}}\right)$ $\quad \forall\, y_{\max} = \sqrt{R_{\max}^2 - x_{\min}^2}$

14: Set $x_{leg} = x_{min}$ and $y_{leg} = \sqrt{R_{max}^2 - x_{min}^2}$ /* desired coordinates are in zone I.

15: Else If $\left(y_{leg} \le \frac{y_{min}}{x_{max}} \cdot x_{leg} \right) \quad \forall\, x_{max} = \sqrt{R_{max}^2 - y_{min}^2}$

16: Set $y_{leg} = y_{min}$ and $x_{leg} = \sqrt{R_{max}^2 - y_{min}^2}$ /* desired coordinates are in zone II.

17: Else set $ratio = \sqrt{\frac{R_{max}^2}{x_{leg}^2 + y_{leg}^2}} \Rightarrow x_{leg} = x_{leg} \cdot ratio, \quad y_{leg} = y_{leg} \cdot ratio$.

/* desired coordinates are in zone V.

18: Grip leg *i*'s claws in position $\left(x_{leg} \quad y_{leg} \quad z_{leg} \right)^T$

The gripping of the claws is being made in a series of actions. Let F_{grab} be the minimum force acting on the leg indicating a successful grip. First, torques are decreased and redefined to the actuators to prevent undesired forces pushing the robot in unwanted directions. Then, the leg is being moved to the desired position $\left(x_{leg} \quad y_{leg} \right)^T$ calculated in algorithm 1. At this location, the robot will move its leg toward the wall (to z_{leg} coordinate) while checking contact forces acting on the leg. When large forces act on the leg in –z direction (normal to the wall), indication of contact with the wall is attained. Then, using inverse kinematics, the robot will move his leg in direction to the floor. It will keep doing so until the forces on the leg will be greater than F_{grab} (successful grip) or until the leg will pass a predefined distance indicating failure to grip. If failure to grasp the desired point x_{leg}, y_{leg} occurs, the robot keeps trying to grip points on a spiral path around x_{leg}, y_{leg} until it succeeds. We assume the leg eventually succeeds grasping. If it succeeds to grasp after multiple trials, the grasping position may be far from the desired one. This may cause a special leg configuration which may prevent the robot from advancing. However, as mentioned, there is no a priori knowledge on the texture of the surface, hence such assumption is inevitable.

4. Implementation and Experimental results

For implementing the model presented above, we used the BIOLOID robotic kit [24]. 16 AX-12+ Dynamixel actuators were used. These actuators are modular DC servo motors containing built-in controllers, drivers, communication protocols and reduction gears. When supplied recommended voltage of 9.6V, the maximum actuator torque is 16.5 kgf·cm and the maximum angular velocity is 51 rpm. Actuator angles and speed can be controlled in a 1024-step resolution. The built-in controller is able to measure actuator angle, speed and torque. This feedback capability is essential for implementing the algorithm. The robots overall length when nonoperational and fully stretched is 750 mm. With an external power source, the CLIBO prototype weighs 2 kg, which makes it very compact and easy to carry. Table 1 summarizes the robots physical properties and other specifications.

The payload of CLIBO is derived from the ability of the actuators and the holding limit of the grippers. Each gripper is capable of holding up to 2 kg of weight. However, According to the equilibrium analysis described in section 2.3 and the maximum torque of the actuators, each leg can hold up to 1.6 kg. Therefore, the payload of CLIBO is about 5 kg, presuming that at least three legs are attached to the wall at any given time. In practice we believe this estimation is too optimistic, and the actual payload would be of about 2 kg. However, payload carrying capabilities were not experimentally verified at this stage.

Weight (unloaded)	2 kg
Overall length	0.75 m
Payload	2 kg
Climbing velocity	12 cm/min
Actuators torque limit	16.5 kgf·cm
Actuators max angular speed	51 rpm
Voltage	9.6V

Table 1 – CLIBO physical properties and specifications.

Because this robot is a prototype for the proof of concept, at this stage the control of the robot was made off-board on a PC. Moreover, an external power supply was wired to CLIBO. With the USB2Dynamixel component, which provides the possibility of communicating with the actuators using a laptop computer, we were able to program the kinematics and equilibrium equations and the presented locomotion algorithm. Using MATLAB we wrote a communication program that makes it possible to read and write data packages to the AX-12 actuators. MATLAB programming was used to write code for the implementation of the locomotion algorithm. CLIBO is controlled by a graphical user interface (Figure 14 that gives the user an ability to input the desired path. It is also used to control certain arguments such as: *Z* (robot's distance from the wall), speed of the robot and the robot's position.

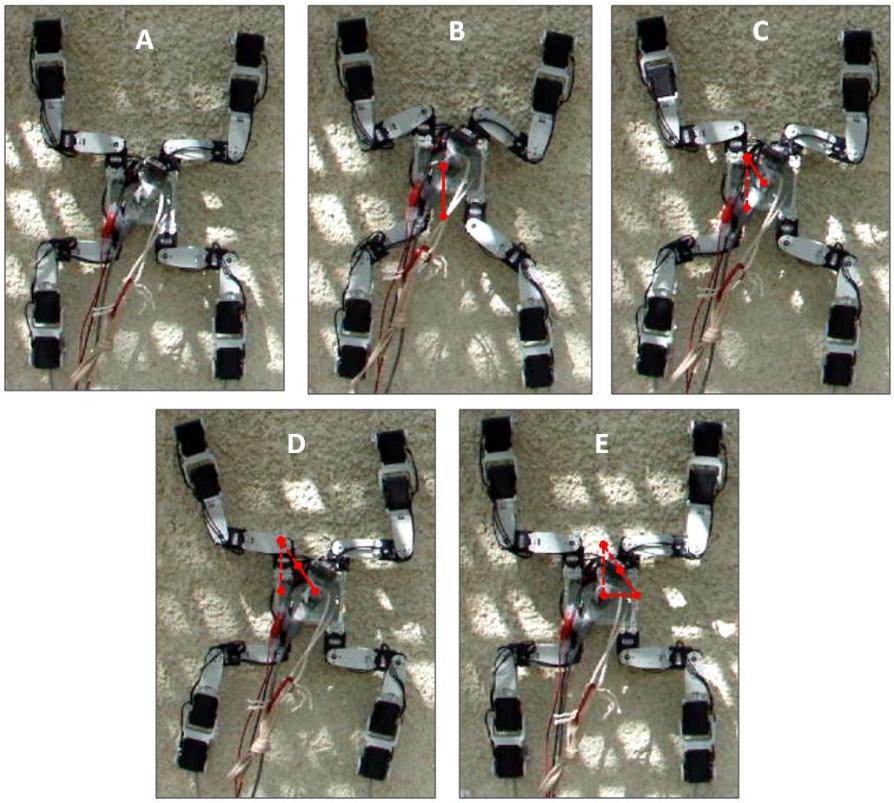


Figure 10: Central body motion in a triangular path, duration time of 31 sec.

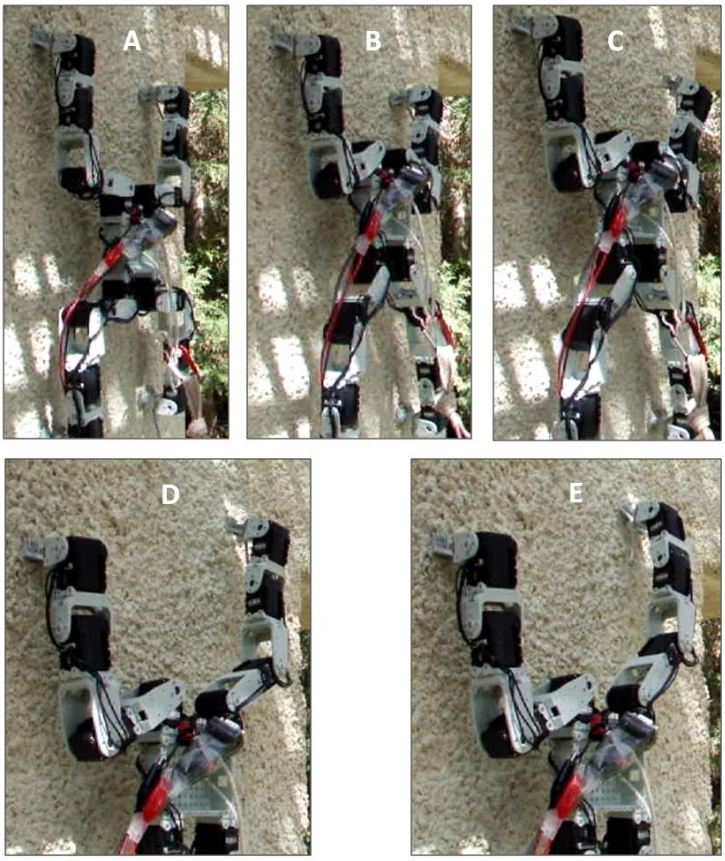


Figure 11: Reposition of the upper-right leg, duration time of 66 sec.

Figure 12 - End Effector/wall interaction.

The robot was tested on a very rough outdoor surface. As mentioned, CLIBO is a prototype built for the proof of concept. While testing the robot, some difficulties due to software and hardware capabilities were encountered. After fine tuning, the robot was able to climb smoothly but very slow. Some snapshots of multiple movements are presented to illustrate its motion. Snapshots of a central body motion test, moving the central body along a triangular path, are shown in Figure 10. In this example, the robot was given a simple three point path to move through. However, the robot divided it to smaller increments according to which the robot checked the position of its legs. It can be seen that the hypotenuse of the triangle was divided into 2 segments. Figure 11 shows the movement of the central body up as much as it can (snapshot A to B). Then, as the legs exceeded their allowed space, the upper-right leg is the first to be repositioned. First, the leg is detached from the wall by moving the EE up toward the ceiling and then away from the wall. In the next step, the leg is moved to its next position only for *x,y* coordinates. The final step is the gripping act; the EE approaches the wall up to the *z* coordinate and then moves down toward the floor until gripping force is obtained. In Figure 12 the interaction between the EE and the stucco wall can be seen. Each claw has independent capability to grasp grooves in the wall. Figure 13 show an experiment result of the robot's central body and one leg. The central body moves quite accurately along the discrete path. At the same time, the leg moves along a same route but parallel to the one the central body is moving along.

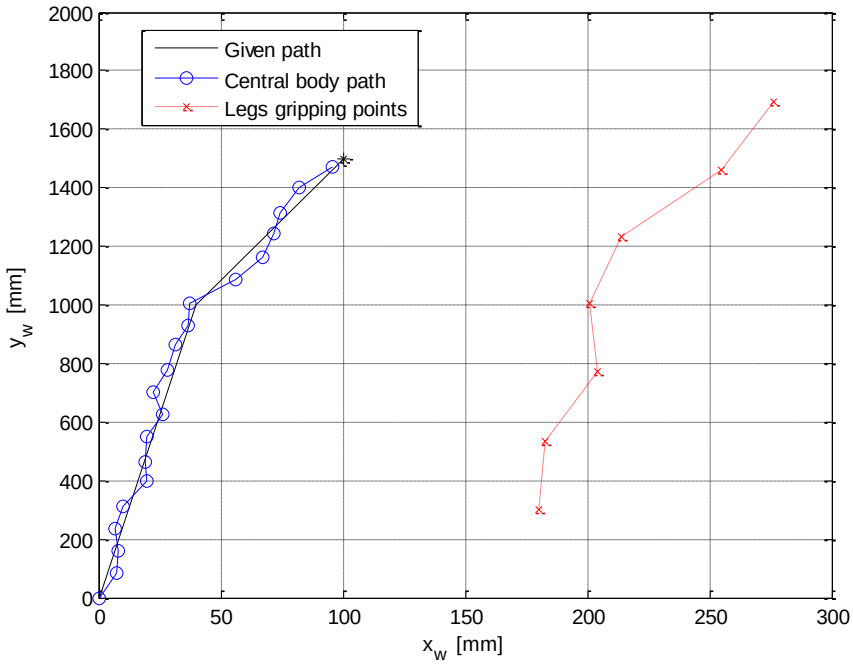


Figure 13: Central body & the upper-right leg's motion along a given path.

During the testing we encountered two problems. One problem regards the stability of the robot on the wall. Faster movements created unwanted dynamical forces. The detachment speed of the leg from the wall was initially set to 0.1m/s, creating dynamical forces dropping the robot. To deal with this matter, an empirical optimization method was used to find an optimal actuator speed for the climbing, decreasing it to 0.06m/s. The second problem, slow communication, has still not been overcome and long delays between moving actions gives the CLIBO a progress velocity of 12 cm/min along the path. Despite these two difficulties, the construction of the gripping device has been well-demonstrated and proved its effectiveness in providing the robot with good attachment reliability.

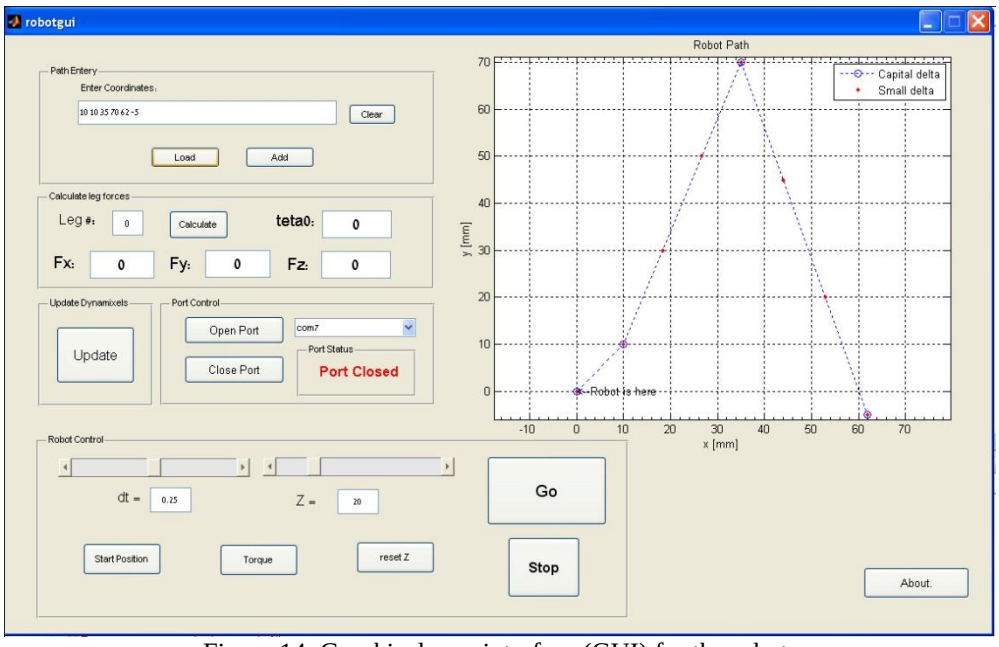


Figure 14: Graphical user interface (GUI) for the robot.

5. Conclusions

This paper presents a four-legged robot, CLIBO, able to climb on rough surface walls using claws attached to its legs. The ability of the CLIBO to climb on these surfaces using a locomotion algorithm was experimentally demonstrated. Several other aspects remain to be further investigated in the future as follows: dynamic locomotion; developing transition algorithms; providing the robot with the ability to stay statically hanging on the wall with no energy consumption; improving on-board computing speed; and providing on-board energy source. Allowing the gripper to attach onto curved surface and adjust the algorithm to uneven terrain would expand the robots maneuverability to more than just vertical walls. Moreover, designing a variety of other grippers involving suction cups or magnets can widen the surfaces CLIBO can climb. Despite the necessity of addressing these issues, CLIBO completed its designed tasks in a reliable way.

Acknowledgment

The authors wish to thank the Paul Ivanier center for Robotics Research, and Production Management, and the The Pearlstone Center for Aeronautical Engineering Studies for their support.

References

[1] K. Daltorio, A. Horchler, S. Gorb, R. Ritzmann and R. Quinn, A Small Wall-Walking Robot with Compliant, Adhesive Feet. IEEE/RSJ International Conference on Intelligent Robots and Systems (2005).

[2] S. Kim, Stickybot, Biommimetic World. Stanford University (2005).

[3] O. Unver, A. Uneri, A. Aydemir and M. Sitti, Geckobot: A Gecko Inspired Climbing Robot Using Elastomer Adhesives. IEEE International Conference on Robotics and Automation, Orlando, Florida (2006).

[4] M.P. Murphy and M. Sitti, Waalbot: An Agile Small-Scale Wall-Climbing RobotUtilizing Dry Elastomer Adhesives. IEEE/ASME Transactions on Mechatronics, Vol. 12, No. 3, June 2007.

[5] M. Greuter, G. Shah, G. Caprari, F. Tache, R. Siegwart and M.Sitti, Toward micro wall-climbing robots using biomimetic fibrillar adhesives. In Proc. of the International Symposium on Autonomous Minirobots for Research and Edutainment (AMIRE'2005), Fukui, Japan (2005).

[6] Clarifying Technologies Inc. (2005). Clarifying Climber III.

[7] X. Chen, M. Wager, M. Nayyerloo, W. Wang and J.G. Chase, A Novel Wall Climbing Robot Based on Bernoulli Effect. IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications, 2008.

[8] L. Briones, P. Bustamante and M. Serna, ROBICEN: A wall-climbing pneumatic robot for inspection in nuclear power plants. Robotics and Computer-Integrated Manufacturing 11, (1994) 287-292.

[9] A. Nagakubo and S. Hirose, Walking and running of the quadruped wall-climbing robot. Proceedings of the IEEE International Conference on Robotics and Automation, (1994) 1005-1012.

[10] R.T. Pack, J.L. Christopher, and K. Kawamura, A Rubbertuator-Based Structure-Climbing Inspection Robot. Proceedings of the IEEE International Conference on Robotics and Automation, (1997) 1869-1874.

[11] L. Briones, P. Bustamante and M.A. Serna, Wall-climbing robot for inspection in nuclear power plants. IEEE International conference on Robotics on Automation, pp. 1409-1411, 1994.

[12] H. Kim, D. Kim, H. Yang, K. Lee, K. Seo, D. Chang and J. Kim, Development of a wall-climbing robot using a tracked wheel mechanism. Journal of Mechanical Science and Technology, 22, 1490-1948, 2008.

[13] A. Albagul, A. Asseni1 and O. Khalifa, Wall Climbing Robot: Mechanical Design and Implementation. Proceedings of the 5th WSEAS international conference on Circuits, systems, signal and telecommunications, 2011.

[14] S. Panich, Development of a Wall Climbing Robot. Journal of Computer Science 6 (10): 1185-1188, 2010.

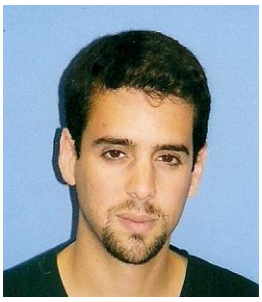
[15] F. Yangqiong and S. Libo, Design and Analysis of Modular Mobile Robot with Magnetic Wheels. WSEAS Trans. On Applied and Theoretical Mechanics, Issue 12, Vol. 3, December 2008.

[16] W. Fischer, F. Tâche and R. Siegwart, Magnetic Wall Climbing Robot for Thin Surfaces with Specific Obstacles. Proceedings of the 6th International Conference on Field and Service Robotics, July 9-12, 2007.

[17] W. Shen, J. Gu and Y. Shen, Proposed Wall Climbing Robot with Permanent Magnetic Tracks for Inspecting Oil Tanks. Proceedings of the IEEE International conference on Mechatronics & Automation, Canada, July 2005.

[18] F. Tâche, W. Fischer, G. Caprari, R. Siegwart, R. Moser and F. Mondada, Magnebike: A magnetic wheeled robot with high mobility for inspecting complex-shaped structures. Journal of Field Robotics, vol. 26, pp. 453-476, 2009.
[19] J.C. Grieco, M. Prieto, M. Armada, and P.G. De Santos, A Six-Legged climbing Robot for High Payloads. Proceedings of the IEEE International conference on Control Applications, Italy, Sep. 1998.
[20] T. Bretl, S.M. Rock, J.C. Latombe, B. Kennedy and H. Aghazarian, Free-Climbing with a Multi-Use Robot. 9th Int. Symp. on Experimental Robotics, Singapore, (2004) 18-21.
[21] M.J. Spenko, G. Haynes, J.A. Saunders, M.R. Cutkosky and A. Rizzi, Biologically Inspired Climbing with Hexapedal Robot. Journal of Field Robotics, Vol 25, Issue 4-5, April – May, 2008.
[22] S. Jensen-Segal, S. Virost, W.R. Provancher, ROCR: Energy Efficient Vertical Wall Climbing with a Pendular Two-Link Mass-Shifting Robot. Presented at the International Conference on Robotics and Automation (ICRA), Pasadena, CA, USA, (2008) 19–23.
[23] A. Asbeck, S. Kim, M. Cutkosky, W. Provancher and M. Lanzetta, Scaling hard vertical surfaces with compliant microspine arrays. International Journal of Robotics Research, 25(12), 1165–1179, 2006.
[24] Robotis, Dynamixel AX-12 User's Manual (2006).

Vitae



Avishai Sintov received his B.Sc degree in Mechanical Engineering from the Ben-Gurion University of the Negev, Beer-Sheva, in 2008. He is currently a M.Sc student in Ben-Gurion University of the Negev. His interests include grasp planning algorithms, climbing robots and mechanical design.

Tomer Avramovich received his B.Sc degree in Mechanical Engineering from the Ben-Gurion University of the Negev, Beer-Sheva, in 2008. He currently works as an engineer at BAE Systems – Rokar, Jerusalem. His occupations include mechanical design, electrical packaging and production engineering.

Amir Shapiro received the B.Sc., M.Sc., and Ph.D. degrees in Mechanical Engineering from the Technion, Israel Institute of Technology, Haifa, in 1997, 2000, and 2004 respectively. Currently he is a Senior Lecturer and director of the robotics laboratory at the Department of Mechanical Engineering of Ben-Gurion University of the Negev, Beer-Sheva, Israel. On 2005-2006 he was a post doctoral fellow at the Robotics Institute of Carnegie Mellon University, Pittsburgh, PA. On the summers of 2007 and 2008 he was a visiting researcher at Caltech – California Institute of Technology. His interests include locomotion of multi-limbed mechanisms in unstructured complex environments, motion planning algorithms for multi-limbed robots, robot grasping-design, control, and stability analysis, climbing robots, snake like robots, multi-robot on-line motion planning, and agriculture robotics.